CSCI-UA.0453-001: Theory of Computation

Problem Set 1

For this problem set, the alphabet is $\Sigma = \{0, 1\}$ unless otherwise specified. All problems are worth 10 points.

Collaboration is allowed, but you must write your own solutions. Write the names of your collaborators (and your own!).

Unless stated otherwise, you must show all intermediate steps and give proper justification or proof.

Problem 1

Draw diagrams of DFAs recognizing the following languages. No justification needed.

1. $\{w : w$ begins with a 0 and ends with a 1$\}$.
2. $\{w : w$ contains at least two 1's$\}$.
3. $\{w : w$ has an even number of 0's and an odd number of 1's$\}$.

Problem 2

Draw diagrams of NFAs recognizing the following languages. Justification needed for the third part.

1. $\{w : w$ contains the substring 0110$\}$.
2. $\{w : w$ starts with 0 and has even length, or starts with 1 and has odd length$\}$.
3. The star closure of $\{w : w$ is any string but 11 and 111$\}$.

Problem 3

Exercise 1.16 on page 86 of Sipser (same in 3rd and 2nd ed).
Note: You can use the procedure to convert an NFA to an equivalent DFA that I described in class; it is a bit different from the one in the book. You must show intermediate steps if any.

Problem 4

Give a regular expression for each of the following languages. Justification (and intermediate steps if any) needed for the third part.

1. $\{w :$ The length of w is a multiple of 3$\}$.
2. $\{w : w$ either starts with 01 or ends with 10$\}$.
3. $\{w : w$ does not contain the substring 001$\}$.

Problem 5

Exercise 1.21 on page 86 of Sipser (same in 3rd and 2nd ed). You must show intermediate steps if any.

Problem 6

The procedure for converting an NFA to an equivalent DFA given in class yields an exponential blowup in the number of states. That is, if the original NFA has n states, then the resulting DFA has 2^n states. In this problem, you will show that such an exponential blowup is necessary in the worst case.

Define $L_n = \{w : \text{The } n\text{th symbol from the right is } 1\}$.

1. Give an NFA with $n + 1$ states that recognizes L_n.
2. Prove that any DFA with fewer than 2^n states cannot recognize L_n.

   Hint: Let M be any DFA with fewer than 2^n states. Start by showing that there exist two different strings, $x \neq y$, $|x| = |y| = n$, that drive M to the same state (by the Pigeon-Hole Principle). Then argue that the strings xz and yz, for any string z, must also drive the DFA to the same state.